Transportation Research Part B 53 (2013) 45–63

Contents lists available at SciVerse ScienceDirect

Transportation Research Part B

journal homepage: www.elsevier.com/locate/trb

Experienced travel time prediction for congested freeways

Mehmet Yildirimoglu, Nikolas Geroliminis *

School of Architecture, Civil and Environmental Engineering, Urban Transport Systems Laboratory, École Polytechnique Fédérale de Lausanne (EPFL), Switzerland

ARTICLE INFO

Article history:
Received 21 July 2012
Received in revised form 21 March 2013
Accepted 21 March 2013

Keywords:
Congestion maps
Travel times
Freeway
Prediction
Traffic flow

ABSTRACT

Travel time is an important performance measure for transportation systems, and dissemination of travel time information can help travelers make reliable travel decisions such as route choice or departure time. Since the traffic data collected in real time reflects the past or current conditions on the roadway, a predictive travel time methodology should be used to obtain the information to be disseminated. However, an important part of the literature either uses instantaneous travel time assumption, and sums the travel time of roadway segments at the starting time of the trip, or uses statistical forecasting algorithms to predict the future travel time. This study benefits from the available traffic flow fundamentals (e.g. shockwave analysis and bottleneck identification), and makes use of both historical and real time traffic information to provide travel time prediction. The methodological framework of this approach sequentially includes a bottleneck identification algorithm, clustering of traffic data in traffic regimes with similar characteristics, development of stochastic congestion maps for clustered data and an online congestion search algorithm, which combines historical data analysis and real-time data to predict experienced travel times at the starting time of the trip. The experimental results based on the loop detector data on Californian freeways indicate that the proposed method provides promising travel time predictions under varying traffic conditions.

1. Introduction

Predictive travel time is valuable information required by drivers and transportation managers to improve the quality of travel and to make control decisions. The provision of travel time information through Advanced Traveler Information Systems (ATISs) enables drivers to make decisions, such as route choice and departure time. In addition, besides the fundamental relation with traffic flow modeling, travel time can be used by transportation agencies to deploy efficient control measures and to prevent potential traffic congestion. Apart from its direct implementation for users and practitioners, travel time experiences strong fluctuations and stochastic traffic phenomena that make its reliable estimation and prediction a challenging physical and mathematical task. Thus, its modeling and estimation requires a combination of correct physics and strong statistical tools.

There are two general methods for obtaining travel time; direct measurement and estimation (Yeon et al., 2008). Direct measurement of travel time can be obtained through test vehicles, license plate matching techniques (automatic vehicle identification, AVI) and ITS probe vehicle techniques. Direct measurement techniques may be misleading in the case of low sampling rates and existence of outlier travel time observations. In order to suppress noise signals, Dion and Rakha (2006) developed an adaptive filtering algorithm, which adjusts its validity window by tracking average travel times. On the other hand, travel time estimation is conducted using the data taken from loop detectors, smart phones or global

* Corresponding author. Tel.: +41 21 69 32481; fax: +41 21 69 35060.
E-mail addresses: mehmet.yildirimoglu@epfl.ch (M. Yildirimoglu), nikolas.geroliminis@epfl.ch (N. Geroliminis).

0191-2615/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.trb.2013.03.006

positioning system (GPS) devices. As numerous freeways around the world are equipped with loop detectors that collect flow, speed and occupancy information, a vast literature of travel time estimation in freeways relies on them. Travel time estimation can be either based on local velocity measurements, or more sophisticated models that attempt to correlate vehicle observations at multiple locations (Coifman, 2002; Coifman and Krishnamurthy, 2007). In addition, estimation models tend to underestimate travel times under congested conditions because of the queue dynamics which cannot be adequately represented in the model. To address this problem, Yeon et al. (2008) made use of discrete time Markov Chains, where the states correspond to whether or not a link is congested, and computed the expected route travel time for several adjacent short links. Furthermore, GPS data provide new opportunities for traffic state estimation and they can be incorporated in estimation algorithms for travel time (Herrera and Bayen, 2010). Mazaré et al. (2012), using the experimental probe data from a field experiment and loop detector data from California Performance Measurement System (PeMS), evaluated the trade-offs between the two types of data. To produce an improved estimate of velocity field, speed measurements from GPS or loop detectors are combined using a mathematical traffic model equivalent to Cell Transmission Model and a traffic state estimation algorithm, the ensemble Kalman filtering. Resulting velocity fields are used to compute travel time, assuming that a vehicle travels at the mean speed reported in each cell. However, the essential problem with travel time information is that it always has to refer to future conditions in the roadway. On the contrary, traffic data collected in real time reflect past or current conditions in the roadway.

Using traffic speed information, there are two ways to compute travel time; instantaneous and experienced. Instantaneous travel time is calculated combining the speed measurements in different locations at the departure time of a trip. On the other hand, experienced travel time is calculated by traveling a trajectory through the velocity field. The time it takes to traverse each segment is calculated, and the speed measurement at the time when the trajectory reaches the next segment is used to compute its travel time. Mathematically speaking, if a freeway is divided into $i = 1,\ldots,I$ sections (I is the most downstream section), and $\tau_i(t_d)$ is the travel time of section i for starting time t_d, then the instantaneous $T^{in}_{S,i}(t_d)$ and experienced, $T^{ex}_{S,I}(t_d)$ travel times to traverse all sections between S and I for departure time t_d are estimated as follows ($T^{ex}_{S,I} = 0,\ \text{for } S \geq I$):

$$T^{in}_{S,I}(t_d) = \sum_{i=S}^{I-1} \tau_i(t_d) \tag{1a}$$

$$T^{ex}_{S,I}(t_d) = \sum_{i=S}^{I-1} \tau_i\left(t_d + T^{ex}_{S,i}(t_d)\right) \tag{1b}$$

To further motivate this research direction, a speed contour plot is presented for a section in freeway I-5S in California in Fig. 1. This plot is constructed with loop detector data for a congested *Friday* of 2011. A few active bottlenecks can be seen in the site that start at different times and propagate upstream. Travel trajectories for instantaneous and experienced travel time approaches are constructed using the speed measurements at the fixed detectors. Space–time (x,t) points on the trajectories are calculated using Eq. (1) and replacing I and S with the corresponding section numbers. Fig. 1 clearly shows

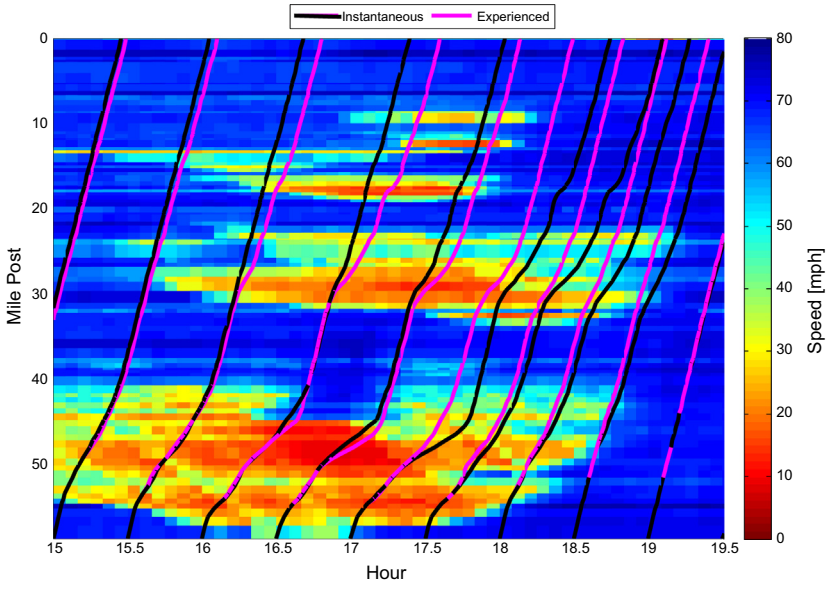

Fig. 1. Speed contour plot and trajectories for a congested day (15:00–19:30) on I5-S freeway.

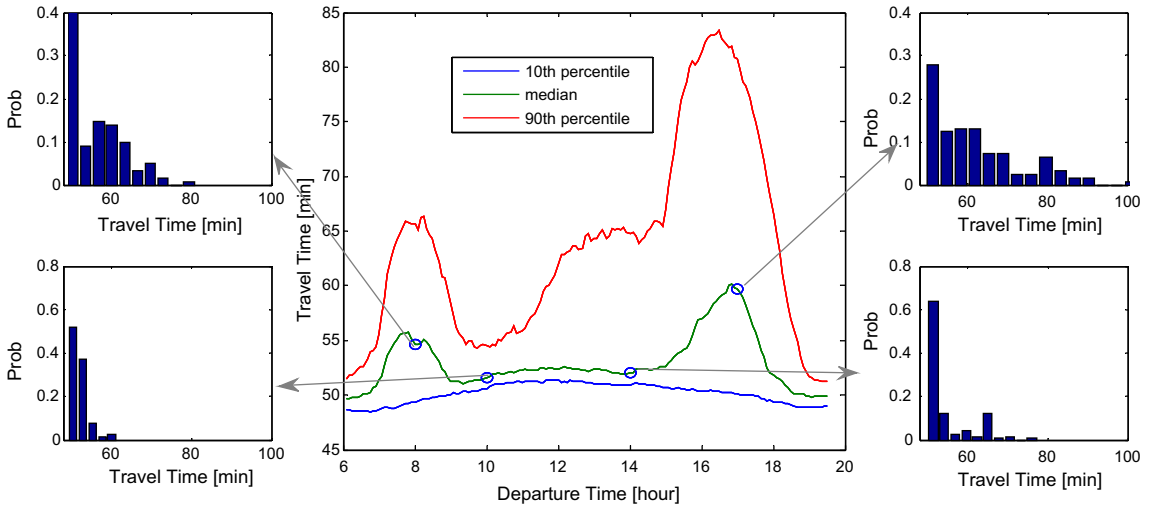

Fig. 2. Median, 10% and 90% percentiles of experienced travel times for '*Tuesday–Thursday*' set in 2011 for different starting times of a trip for a 60 mile section in I5-S freeway.

the difference between instantaneous and experienced travel time by plotting a few vehicle trajectories for the two estimators. Note that these differences can be quite significant especially during the congestion onset and dissipation. This indicates that estimation of travel time should not be solely based on the traffic data collected in real time, but also the future recurrent traffic conditions should be integrated from historical data.

Nevertheless, simple historical average provides large estimation errors, which are due to the stochastic characteristics of traffic especially under congested conditions (formation of queues, demand uncertainty, etc.). To emphasize more the high variation of travel times from day to day and the high error of historical average as an estimator, Fig. 2 plots for different departure times, the median, 10th and 90th percentile of experienced travel time for all '*Tuesdays*', '*Wednesdays*' and '*Thursdays*' of 1 year in the same study site. From the time series graph, it is clear that even uncongested off-peak periods according to the median travel time value in the early afternoon have some significant probability to experience strong delays. Distributions for four different departure times are illustrated as well. Thus, there is a need for development of an accurate short-term traffic state prediction to be integrated in the estimation of experienced travel times.

The need for short-term traffic prediction led to the development of various forecasting algorithms. These methods can be broadly classified in two major categories; parametric methods (e.g. linear regression (Zhang and Rice, 2003), time series models (Yang, 2005; Min and Wynter, 2011), Kalman filtering (Okutani and Stephanedes, 1984; Van Lint, 2008)) and non- parametric methods (neural network models (Ledoux, 1997; Vlahogianni et al., 2005; Van Lint, 2006), support vector regression (Vanajakshi and Rilett, 2007), simulation models (Liu et al., 2006)). In the past years, neural network models have gained attention in transportation field and are frequently applied in traffic state prediction. The majority of transportation applications of neural networks is based on simple back-propagation algorithm (e.g. Adeli, 2001). However, other computing algorithms such as counter-propagation neural networks have the potential to improve prediction results (Dharia and Adeli, 2003). Additionally, many previous studies utilized macroscopic traffic flow models along with Kalman filtering technique to predict traffic states (Nanthawichit et al., 2003; Wang et al., 2008). Moreover, Fei et al. (2011), with a Bayesian framework, and Du et al. (2012), with an information fusion model, attempt to predict short-term travel time distribution considering the fact that the mean of a short-term travel time distribution may not be an accurate tracking indicator.

Models where travel time is directly used as the state variable, suffer from the fact that travel time in the previous time interval is needed to predict future travel time. However, in practice, trip travel time is usually greater than the prediction interval. Therefore, experienced travel time in the previous interval cannot be used to calculate future travel times. On the other hand, instantaneous travel time, which does not consider congestion or speed evolution, is available at the departure time to predict instantaneous travel time in the following time interval. Data-driven approaches, which make use of instantaneous travel time, are consistent under some cases with the transitional physics of traffic flow and they are capable of constructing the underlying behavior of traffic without strong assumptions on its temporal evolution (see for example Jiang and Adeli, 2004; Vlahogianni et al., 2005). However, data-driven approaches cannot explicitly infer knowledge from point measurements for estimating link performance measures (Vlahogianni et al., 2008). This is because spatiotemporal traffic flow dynamics are mainly governed by the queue formation and dissipation at point bottlenecks. Abrupt changes of traffic phenomena (e.g. lane changes, capacity drop, merge behavior, oscillations) can affect congestion development and propagation in various ways that require physical than statistical models to be explained (see for example Leclercq et al., 2011; Li et al., 2010; Treiber et al., 2010). These characteristics of traffic's transitional behavior and the existence of variant traffic regimes may not be identified by statistically-oriented or data-driven approaches and increase their estimation and prediction errors.

In contrast to the aforementioned existing methodologies, the approach presented in this paper benefits from the available traffic flow essentials (e.g. shockwaves and bottlenecks). The proposed method makes use of both historical and real time traffic information to provide travel time prediction. Instead of identifying traffic flow patterns using statistical methods (that sometimes might not succeed to capture complex phenomena of traffic flow), we propose to integrate in the methodology, identification of traffic patterns with traffic flow theory fundamentals, for example with shockwave analysis and bottleneck identification. The aim of the proposed model is to respond to changes in traffic pattern in real-time and to provide the expected 'experienced' travel time reflecting both a priori knowledge (i.e. historical dataset) and real-time traffic data through an incremental learning approach.

Considering the lag associated with experienced travel time in real-time information, this paper focuses on congestion evolution to develop spatiotemporal traffic state maps and to construct predicted travel trajectories on them, leading eventually to predicted experienced travel time and the travel trajectory. The current and historical speed data are utilized in the prediction framework. However, due to the strong variability and difficulty in predicting them, speed data is first processed through an algorithm that would identify congested space–time domains. This approach, therefore, prefers to consider congestion evolution rather than speed evolution, because of the ease in detecting congestion evolution pattern in real time. An existing bottleneck identification algorithm is utilized to determine the location and spatial extent of the bottlenecks (Chen et al., 2004). The algorithm is used in this study to store the major traffic events likely to be observed on the roadway (in historical data) and to track real-time conditions (in current data). Using the shockwave phenomena and identified bottleneck locations in real-time, the impact of a bottleneck can be predicted before it completely develops. Historical information can be useful to determine the characteristics of the bottlenecks (i.e. spatial extent and duration) and so, predict their impacts. Nevertheless, as we will show later, traffic conditions significantly vary from day to day (even for similar demand conditions) and as a result the size of a bottleneck in the space–time domain and travel speed of vehicles in this domain experience strong fluctuations. Hence, a simple prediction based on historical average or a partitioning of traffic conditions based on days (weekdays–weekends) or times of day (AM or PM peak) might introduce significant estimation errors.

This study partitions the historical dataset in clusters with similar characteristics based on the traffic patterns observed in the roadway. The building block of the methodology is the development of stochastic congestion maps, which identify the probability that a space–time domain is congested. This probability might have strong fluctuations for days with significantly different level of congestion and as a result it can decrease the performance of the prediction. Hence, this study develops a congestion search algorithm to revise the state of a priori knowledge according to the newly available information in real-time. Finally, the proposed approach develops a speed profile to construct travel trajectories using the predicted congestion evolution pattern and to calculate expected experienced travel time downstream for a given starting trip time.

The remainder of the paper is organized as follows: Section 2 develops the methodological framework, Sections 3 and 4 provide an implementation of the methodology in a real case study by utilizing 1 year of data for a congested Californian freeway (I5-S), while Section 5 includes discussion and future directions.

2. Methodology

The methodological framework of this approach includes (i) an existing bottleneck identification algorithm, (ii) clustering of data in traffic regimes with similar characteristics, (iii) development of stochastic congestion maps, (iv) an online congestion search algorithm, which combines historical and real-time data, and (v) a simple speed profile. While the proposed methodology uses loop detector data, it is not constrained to other sources of data (e.g. GPS data), given that this data can be utilized for bottleneck identification.

2.1. Bottleneck identification algorithm: a review

Chen et al. (2004) developed an algorithm to automatically identify bottleneck locations, their activation and deactivation times, and their spatial extents using loop detector data and focusing on speed measurements. Our methodology, which is described in the following sections, is not constrained by the specific algorithm. This method compares each pair of adjacent detectors and determines the existence of bottleneck when

- Speed difference between upstream and downstream detectors is above the minimum speed differential, Δv_{min} threshold.
- Speed at upstream detector is below the maximum speed threshold, v_{max}.

Chen et al. (2004) choose values of v_{max} = 40 mph and Δv_{min} = 20 mph with data aggregated at 5 min intervals taken from California freeways. These parameters may need to be adjusted depending on the application. Wieczorek et al. (2010) discusses the effect of parameters on the model results, develops assessment criteria to select the optimal configuration of parameters, and by using the results of the bottleneck identification algorithm, maps recurrent congestion in time and space.

The algorithm has also an offline part to identify the sustained bottleneck locations. This part smoothens the results of the online part, and fills in the small gaps in bottleneck detection at a particular detector. Basically, if several consecutive time periods are identified as bottleneck points, but one point in the middle failed to be identified so, offline part fills in this gap. Considering the fact that deactivation and reactivation of a bottleneck is not possible in such a short period of time, offline

part accounts for the problems that could arise from the selection of parameters or missing data. The congested region affected by an active bottleneck can be defined using the speed measurements at upstream locations. This description is slightly modified in our estimation from the original algorithm. A congested region associated with a bottleneck ends at the detector location where two consecutive upstream detectors have more than v_{max}, while a single detector with more than v_{max} is sufficient to enclose the congested region in the original algorithm. We note that with this alternation, the methodology provides better identification especially in the offset of congestion.

Identification of congested sections in an automated way allows to restore the major traffic events that occur on the roadway (in historical data) and to keep track of traffic conditions in real time (in current data). However, since the algorithm has an offline part, it is not possible to smooth the results in real-time. The following methodological parts are not constrained to the specific bottleneck identification algorithm or data. The choice was based on the small computational effort, combined with its proved accuracy to estimate congested conditions. Any type of data and algorithm that can provide accurate bottleneck locations and formation of queues in a space–time domain can be directly utilized in the remaining of the paper.

2.2. Clustering of days with similar traffic patterns

Historical traffic patterns are crucial to the development of a travel time prediction framework due to the recurrence of traffic events. To use the historical dataset in a useful and efficient manner, days with similar traffic patterns (i.e. speed profiles) should be identified to decrease the randomness of traffic conditions. Otherwise large variations and temporal bias might be experienced by utilizing very heterogeneous data. Clustering techniques have been already used in transportation field to analyze traffic flow patterns, see for example Weijermars and Van Berkum (2005) or Ji and Geroliminis (2012). Since travel times are computed using local velocity measurements in this study, time-dependent speed measurements along the roadway can be used in the clustering step. Note that traffic speed data is processed to identify congested space–time domains, and the current and historical congestion information is the input to travel time prediction framework. Considering the high variability in traffic speed data and the difficulty in predicting them, our methodology mainly focuses on congestion evolution rather than speed evolution. In other words, this study uses a binary approach (i.e. congested or free-flow) to predict traffic states. However, as identification of traffic patterns involves only the use of historical data, this binary approach is not needed in clustering; traffic speed data can be directly used in the clustering step. Without clustering the variance of travel time for a given departure time is significantly larger and this has a direct erroneous effect in the prediction.

Time-dependent local velocity measurements at different locations and multiple days are the input to this step. Each variable in the input dataset represents a velocity measurement for a particular time period and a particular roadway section. Since a high number of sections on the roadway and time periods in a day lead to a large number of variables (e.g. "180 time periods per day" × "89 roadway sections" = 16,020 variables for the study site of the paper), it is not straightforward to define a metric to compare and cluster days with strong similarity for the freeway route under consideration. Reduction in the variable size is required to proceed with clustering. Additionally, results of clustering can be corrupted by the noises in the original data (Milligan, 1980); speed data itself may contain noises which are not useful to explain general trends. Thus, clustering may benefit from a preprocessing step of variable selection which results in de-noising of data. Few leading principal components identified by Principal Component Analysis (PCA) can help us remove the noise in the dataset, reduce the dimensions and so apply more rigorous clustering techniques. First, original dataset is processed through PCA to reduce its dimensions and to remove the noise. Second, resulting principal components are used to cluster the days with similar traffic patterns.

2.2.1. Principal Component Analysis (PCA)

PCA is a well-established technique to reduce the dimensions of the dataset and to compress the data, see for example Nagendra and Khare (2003). PCA, using the orthogonal transformation, converts a set of observations with correlated variables into a set of observations with linearly uncorrelated variables, which are called principal components (PCs). In other words, it transforms the data into a new space which has most of the information (or energy) of the original data, but with a lower dimension. It lists PC's in the descending order of the variance associated with them. With respect to our problem, a freeway route might have detectors installed every a few hundred meters that provide speed and flow measurements every a few minutes. This creates an immense data set that cannot be directly utilized to cluster different days.

Suppose matrix $\boldsymbol{X}_{(m\times n)}$ is the original data set with rows corresponding to observations (e.g. different days) and columns corresponding to variables (e.g. time- and space-dependent local velocity measurements). The variables are correlated, and there exists another set of uncorrelated variables $\boldsymbol{S}_{m\times n}$, which is a linear combination of $\boldsymbol{X}_{m\times n}$;

$$\boldsymbol{S} = \boldsymbol{P} * \boldsymbol{X} \tag{2}$$

where $\boldsymbol{P}$ is the $(m \times m)$ projection matrix. The aim of PCA is to find a projected space $\boldsymbol{S}$ whose covariance matrix is diagonal, or in other words variables are uncorrelated. The first row in the projection matrix represents a new axis in the uncorrelated set, and the resulting values from the first row create a new variable whose variance is the maximum among all possible choices (i.e. first principal component). The second row has the same properties for the set without the first principal component; so on and so forth.

PCA algorithm can be summarized as follows:

1. Subtract from each element the mean value of the corresponding column: $X[j,i] \leftarrow X[j,i] - \mathbb{E}_i[\boldsymbol{X}]$.
2. Compute correlation matrix $\boldsymbol{C} = \boldsymbol{X}^T\boldsymbol{X}$.
3. Compute eigenvalues $|\boldsymbol{C} - \lambda_i \boldsymbol{I}_n| = 0, i = 1, 2.., n$.
4. Compute eigenvectors $\boldsymbol{C}e^i = \lambda_i e^i$.
5. Choose $p < n$ eigenvectors: $e^1, \ldots, e^p$ with $\lambda_1 \geqslant \lambda_2 \geqslant \ldots \geqslant \lambda_p$.
6. Project data into new space $\boldsymbol{S} = \boldsymbol{P} * \boldsymbol{X}$, $\boldsymbol{P} = \begin{bmatrix} e_1^1 & \cdots & e_1^p \\ \vdots & \ddots & \vdots \\ e_n^1 & \cdots & e_n^p \end{bmatrix}$.

The eigenvalues represent the energy or the variance of the dataset along the eigenvector directions. The cumulative energy value e for the nth eigenvector is the sum of the energy across the eigenvectors from 1 to n, which can be formulated as $e[n] = \sum_{i=1}^{n} \lambda_i$. While selecting a subset of eigenvectors as the basis vector, the goal is to choose the minimum number of components that achieves reasonably high value of e on a percentage basis. The number of components p can be determined using the inequality $e[p]/e[n] > \kappa$ and setting an accuracy level κ (e.g. 0.95).

2.2.2. Gaussian Mixture Modeling

After reducing the dimensions of the dataset, Gaussian Mixture Model (GMM) is applied to create clusters in the historical dataset. GMM is the combination of multivariate normal density components, and it estimates normal distribution parameters using expectation maximization (EM) algorithm. GMM is often used for clustering purposes, and unlike other clustering methods, it is not solely based on the distance between the observations, but it is based on the distribution of data points. GMM is a more appropriate method than k-means clustering, when clusters have different sizes and correlation within them, which is the case for traffic data.

Consider now $(m \times p)$ matrix $\boldsymbol{S} = \left\{s_j^i\right\}_{j=1..p}^{i=1..m}$ with rows corresponding to observations (e.g. days) and columns corresponding to the principal components. The probability density function (PDF) of $\boldsymbol{S}$ will be modeled as a mixture of K Gaussian distributions.

$$p(S) = \sum_{k=1}^{K} \alpha_k * p(S|\mu_k, \Sigma_k) \tag{3}$$

PDF of each Gaussian: $p(S|\mu_k, \Sigma_k) = N(\mu_k, \Sigma_k)$, mixing coefficients: $\sum_{k=1}^{K} \alpha_k = 1$, μ_k, Σ_k: mean and covariance matrices of Gaussian $k = 1, \ldots, K$. Probability that the data is explained by Gaussian k: $\alpha_k = p(k) = \sum_{i=1}^{m} p(k|s^i)$.

The parameters of GMM are the means, covariance matrices and mixing coefficients;

$$\Theta = \{\mu_1, \ldots, \mu_k, \quad \Sigma_1, \ldots, \Sigma_k, \quad \alpha_1, \ldots, \alpha_K\} \tag{4}$$

However, the parameters of GMM cannot be directly estimated because of unobserved latent variables. EM is an iterative method to find maximum-likelihood estimates of the parameters, where the model depends on unobserved latent variables (Dempster et al., 1977). EM attempts to find the optimum of the likelihood of the model given the data;

$$\max_{\Theta} L(\Theta|S) = \max_{\Theta} p(S|\Theta)$$

$$\max_{\Theta} p(S|\Theta) = \max_{\Theta} \prod_{i=1}^{m} \sum_{k=1}^{K} \alpha_k * p(s^i|\mu_k, \Sigma_k) \tag{5}$$

$$\max_{\Theta} \log p(S|\Theta) = \max_{\Theta} \sum_{i=1}^{m} \log \left(\sum_{k=1}^{K} \alpha_k * p(s^i|\mu_k, \Sigma_k) \right)$$

At each estimation step l, parameters are updated as;

$$\alpha_k^{(l+1)} = \frac{1}{m} \sum_i p(k|s^i, \Theta^{(l)})$$

$$\mu_k^{(l+1)} = \frac{\sum_i p(k|s^i, \Theta^{(l)}) s^i}{\sum_i p(k|s^i, \Theta^{(l)})} \tag{6}$$

$$\Sigma_k^{(l+1)} = \frac{\sum_i p(k|s^i, \Theta^{(l)})(s^i - \mu_k^{(l+1)})(s^i - \mu_k^{(l+1)})^T}{\sum_i p(k|s^i, \Theta^{(l)})}$$

The parameter set Θ is updated iteratively until the log likelihood is increased by less than a certain threshold value.

Optimal number of clusters (K) can be determined by the use of an average silhouette width (Rousseeuw, 1987) or information measures such as Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC). In addition to the optimal number of clusters, the stability of clustering results is also crucial. GMM, whose initialization is random or based on k-means results, should return the same results every time it is repeated to ensure the accuracy and the robustness of the algorithm. By varying the number of components in clustering, we can identify a value that will create a high silhouette width and robust clusters. Silhouette is a common technique to validate clusters of data; it provides a distinct silhouette width value representing how well each observation belongs to its cluster and how dissimilar it is from the other clusters (Rousseeuw, 1987). Optimal number of clusters can be determined by maximizing the average of silhouette width values in a data set.

2.3. Stochastic congestion maps

To add a probabilistic flavor in the formation of bottlenecks from day to day that will contribute in the accurate prediction of experienced travel times, we introduce a new physical concept of describing spatiotemporal traffic patterns, the *stochastic congestion map*. It represents the likelihood of congestion occurrence at a given space–time point based on the observations for many days of the same cluster in the historical dataset. The probability of observing congestion at roadway segment i at time t is calculated separately for each cluster k.

$$p^k(i,t) = \frac{1}{D_k}\sum_{m=1}^{D_k} f(i,t,m) \tag{7}$$

$$f(i,t,m) = \begin{cases} 1 & \text{if segment } i \text{ is congested at time } t \text{ on day } m \\ 0 & \text{otherwise} \end{cases}$$

where D_k is the number of days in cluster k. Function f is estimated with the bottleneck algorithm of Section 2.1. We consider the tool of stochastic congestion maps appropriate for various research methodologies, such as travel time reliability and predictive control.

Once cluster analysis and bottleneck identification algorithm are applied, a stochastic congestion map showing the probability of congestion occurrence at a particular space–time domain is estimated for each cluster. Two arbitrary clusters are shown in Fig. 3a and c. Each cluster is divided into subsets using certain threshold probability values (e.g. from 0.05 to 1), and

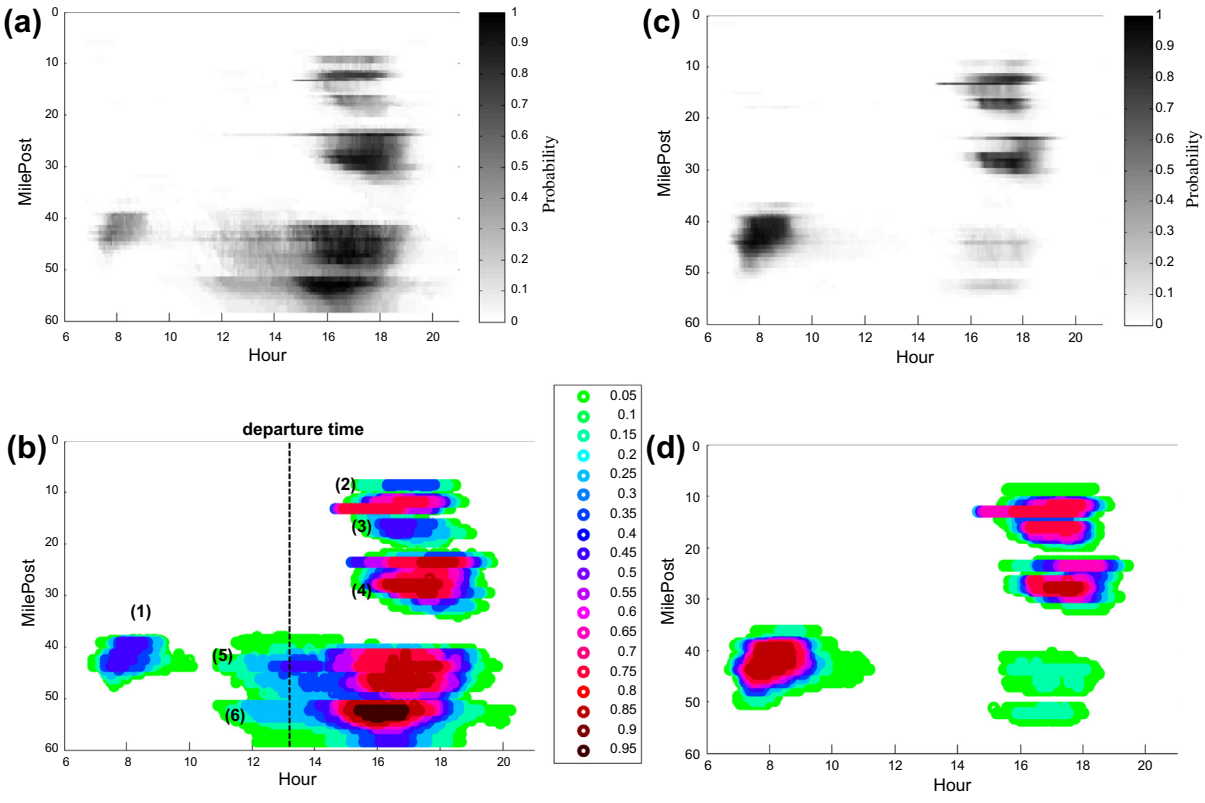

Fig. 3. (a and c) Stochastic congestion map and (b and d) blocks in stochastic congestion map.

blocks corresponding to a certain likelihood of occurrence is created for each probability value. Blocks associated with the lowest threshold are constructed by the congested points with probability greater than 0.05 (i.e. occurring more than 5% of the analyzed days), blocks associated with second lowest threshold is constructed by the points with probability greater than 0.10, and so on so forth (like a cumulative 2D distribution). Thus, heavy congestion observations are associated with low threshold values, while light congestion situations are associated with high threshold values. Fig. 3b and d represents the blocks along with threshold values associated with two clusters having different levels of congestion. Note that the block associated with the lowest threshold is a superset of all the other blocks, the one associated with second lowest is a superset of the blocks associated with third and higher lowest thresholds, so on so forth. In other words, light colored dots (e.g. green dots; block associated with the lowest threshold) in Fig. 3b and d do exist beneath dark colored dots (e.g. any dot darker than green; blocks associated with second and higher lowest thresholds), although they are not visible. In addition, the difference between blocks is roughly in the shape of asymmetric rings around a given core, which can be considered as a separate bottleneck. These results are of great importance to our analysis because they show that even the location and duration of bottlenecks are roughly known a priori, a more careful look identifies strong stochastic phenomena that can vary travel times from one day to the other, which is also observed in Wieczorek et al. (2010). For example, bottleneck #5 in Fig. 3b starts at location with milepost 40, but its extension in time and space varies from day to day. For the remainder of the paper, bottleneck will refer to a varying spatiotemporal congested area caused by a particular active bottleneck location, and block will refer to a bottleneck subset that corresponds to certain probability of occurrence. In Section 3.3, we discuss an alternative method to construct stochastic congestion maps by introducing physical bounds on shockwave speeds.

The proposed approach does not distinguish between recurrent and non-recurrent phenomena because of two reasons. First, the available incident dataset does not allow us to quantify their effects on traffic conditions. Second, traffic prediction in non-recurrent conditions requires a different approach considering incident characteristics, such as the clearance time of the incident, the loss of capacity and others. Newell's 3-detector (Newell, 1993) model can be one of the ways to incorporate non-recurrent conditions in this framework. Based on the vehicle counts at two detector stations, vehicle counts at some intermediate location are computed. However, Newell's model cannot tolerate ramp flows between the two detectors. With respect to our problem, in case of unexpected congestion detection, the 3-detector model can be activated to determine the bottleneck extent. However, such a model would require prediction of incident duration (i.e. duration in which bottleneck is active), prediction of departure curve at the bottleneck location and prediction of arrival curve at downstream of the freeway (right before the closest ramp to the bottleneck location). In our case, non-recurring events are indirectly addressed through stochastic congestion maps with probability values as low as 0.05, which is not very recurrent. For example, if an accident creates much higher delays than recurrent bottlenecks (as this is expressed by low values of thresholds in the stochastic congestion maps), the travel time prediction can integrate some component of this oversized delay. However, note that travel time for a specific departure time can significantly vary even under recurrent conditions. To address the travel time variability and to provide more accurate travel time information, an online congestion search algorithm is developed in this study that allows us to switch between blocks and congestion evolution patterns represented by them.

2.4. Online congestion search algorithm

The aim of the online congestion search algorithm is to provide the connection between real-time and historical information. Binary information (congested or not congested) collected in real-time is compared with the possible bottleneck shapes (i.e. blocks in stochastic congestion map) in the historical database, and the best representing block is selected. Note that stochastic congestion map, which incorporates continuous probability values, is already discretized to construct blocks, which consist of binary congestion data and are comparable with real-time information. Before delving into the details of the algorithm, note that there are multiple cores (bottlenecks) in the congestion map. In addition, although there may be a correlation between the size of the bottlenecks for a given day, it is intuitive that they are not highly dependent, as they occur at different locations and the propagation of the one does not significantly affect the propagation of the others. Therefore, the congestion map is divided into six separate bottlenecks as shown in Fig. 3b, and the congestion search algorithm is applied separately for each of them to select the threshold value that best represents the real-time traffic conditions on the spatiotemporal area covered by the corresponding bottleneck.

The idea behind the online congestion search algorithm is to provide travel times based on the expected traffic conditions at the very beginning and integrate real-time congestion information to specify the shape of the bottleneck in future time periods. If there is no real-time information about the given bottleneck at the departure time, the expected threshold value (i.e. probability of 0.5) is used to compute the predicted travel time. However, if the departure time is later than the starting time of a downstream bottleneck, then we are able to compare real-time traffic information with congestion maps for particular bottlenecks (e.g. for a departure time at 13:00 in Fig. 3b bottlenecks #5 and #6 are already active and thresholds can be updated). The algorithm determines the threshold value or block that would best represent the real-time congestion information obtained till the departure time. The threshold value or block estimated by the congestion search algorithm for each bottleneck is applied to construct the predicted congestion map, and travel time for the given departure time is computed on this predicted congestion map. A graphical representation of the algorithm is provided later.

For a given bottleneck, selection of threshold value is done by the following similarity metric;

$$\arg\max_{p}(g(p,j,k)+h(p,j,k))$$

$$g(p,j,k) = TP\left(c^k_{p,j}, RTI\right) \tag{8}$$

$$h(p,j,k) = TN\left(c^k_{p,j}, RTI, c^k_{1,j}\right)$$

p is the threshold (block) index, j is bottleneck index, k is cluster index, RTI real-time information (the set of congested points observed till the departure time), $c^k_{p,j}$ is set of points defined by bottleneck j, threshold p and cluster k, TP (true positive) is number of correctly classified congested points of RTI by $c^k_{p,j}$, TN (true negative) is the number of correctly classified non-congested points of RTI by $c^k_{p,j}$ in the set defined by $c^k_{1,j}$.

Note that the set of points used to determine TN, is defined by the maximum size that the given bottleneck can get (i.e. block associated with lowest threshold; $c^k_{1,j}$). Maximum size of the bottleneck is needed to identify the basis where number of rightly classified non-congested points can be calculated. Since real-time information is available only till departure time, the points up to the departure time are used in the search mechanism. In addition, a moving time window of 2 h is used to keep track of varying conditions. If the same performance value is computed for multiple threshold values, then the one closest to the expected threshold value is chosen. This indicates that the algorithm always selects the conditions that are most likely to be observed.

2.5. Online cluster switch

Partitioning of historical dataset may give some hints about how to assign the days to the clusters in a predetermined way. If a cluster mainly consists of particular days of the week (e.g. Mondays only), the corresponding days can be pre-assigned to that cluster. However, it is very likely that the cluster might contain other days of the week. Hence, traffic prediction should be done in a flexible way that allows the algorithm to switch between clusters. On the contrary, the choice of cluster should not change at each time step, because it may bring unrealistic fluctuations in travel time. It should be considered as a tactical level decision, while the choice of threshold (block) is an operational level decision.

Therefore, online cluster switch is implemented if the following condition applies;

$$TP\left(c^{alt}_{1,j}, RTI[t-2,t]\right) > \alpha * TP(c^{pre}_{1,j}, RTI[t-2,t]) \tag{9}$$

where $c^{alt}_{1,j}$ and $c^{pre}_{1,j}$ are the smallest blocks of an alternative cluster and the predetermined cluster, respectively. $RTI[t-2,t]$ is the real-time information with a time window of 2 h, $\alpha > 1$ is an empirical coefficient that restricts strong fluctuations in the choice of cluster.

In other words, the algorithm switches to the alternative cluster when the number of congested points identified by the alternative cluster is α times greater than the ones identified by the predetermined cluster. Note that cluster switch is implemented separately for each defined bottleneck, and the sensitivity analysis for the parameter α is presented in Section 4.2.

Note that in Fig. 3b and d, which present the blocks for different clusters, there are many congested points in one cluster that do not exist in the other. For instance, if the predetermined cluster (Fig. 3d) is not able to address for the congested points observed in real-time (e.g. left part of the bottlenecks 5 and 6 in Fig. 3b) and Eq. (9) is satisfied, then the algorithm switches to the alternative cluster. The cluster switch can also address a component of non-recurrent events by switching to a more congested cluster.

2.6. Speed profile

The developed methodology to predict the experienced travel times requires two pieces of future information (i) prediction of congestion development and propagation on the roadway (e.g. bottlenecks) and (ii) prediction of speed profiles. By the methodology described so far, the future speed profile inside and outside the bottleneck space–time domain is still unknown at the beginning of a trip. To estimate the trajectory of a vehicle which runs in the predicted space–time domain of a congestion map, different speeds are considered for congested, V_c, and uncongested conditions, V_f.

Speed information at the departure time can bring valuable insight into this problem and estimate V_c and V_f.

$$V_c = \frac{1}{N_c}\sum_{i=1}^{N_c} v_i^{t_d} \text{ and } V_f = \frac{1}{N_f}\sum_{i=1}^{N_f} v_i^{t_d} \tag{10}$$

where t_d is the departure time, $v_i^{t_d}$ is the speed measurement on roadway segment i at t_d, N_c is the number of sections registered as congested at t_d, N_f is the number of sections registered as uncongested at t_d.

However, the speed profile defined by Eq. (10) does not account for the speed variability along the roadway and it can produce erroneous results, especially when congested speeds vary with different roadway sections. Therefore, the congested speed, V_c, is modified for each roadway segment r as follows. If a particular roadway section r is registered as congested at the departure time, t_d, and if it remains congested in the predicted congestion map, its speed at the time of arrival t_a, estimated by Eq. (1b) for time $[t_d + T^{ex}_{s,r}(t_d)]$. Speed $v_r^{t_a}$ is the average of the congested speed and the speed of the section at the time of departure, i.e. $v_r^{t_a} = \frac{1}{2}(V_c + v_r^{t_d})$. In this way, local congestion phenomena are integrated in the prediction and results are further improved. A more detailed analysis could estimate V_c as a function of the threshold index p for each bottleneck.

3. Case study

For the application of the methodological framework, data from PeMS is used. PeMS collects 30-s loop detector flow and occupancy data throughout the Californian state. Then, it processes them and fills in the missing detector data to compute 5-min flow, occupancy and speed averages (Chen et al., 2001). For this study, a 60 mile section of I-5S in the district of San Diego/Imperial is selected, between mileage 0 and 60. A 20 miles portion of the freeway, along with the detector locations is presented in Fig. 4. Considering the detector quality and the effect of strong recurrent congestion in multiple locations, the selected roadway section is a challenging study to evaluate the methodological framework. 5-min loop detector data is collected through PeMS for the whole year of 2011. The dataset is divided randomly into two parts; Training Set (~80%–289 days) and Testing Set (~20%–76 days).

3.1. Bottleneck identification algorithm

Bottleneck identification algorithm is applied for the training dataset and stochastic congestion maps are estimated in the next steps of the methodology. Fig. 5 presents the speed contour plot for a single day and the congested regions identified by the algorithm. It is clear that the identification is reliable both in the onset and offset of congestion.

3.2. Clustering of days with similar traffic conditions

This approach provides a way to cluster data based on their congestion profile. A qualitative approach (e.g. all Mondays are the same) is not appropriate given the stochastic characteristics of traffic especially under congested conditions (formation of queues, demand and capacity uncertainty, etc.). Clustering is crucial to the prediction performance, because it defines the library on which the learning scheme is implemented.

Before clustering, PCA, as described in Section 2.2.1, is applied to reduce the dimensions of the dataset and to remove the noise. PCA analysis shows that 100 principal components carry 95% of the variance in the original data of 16,020 variables

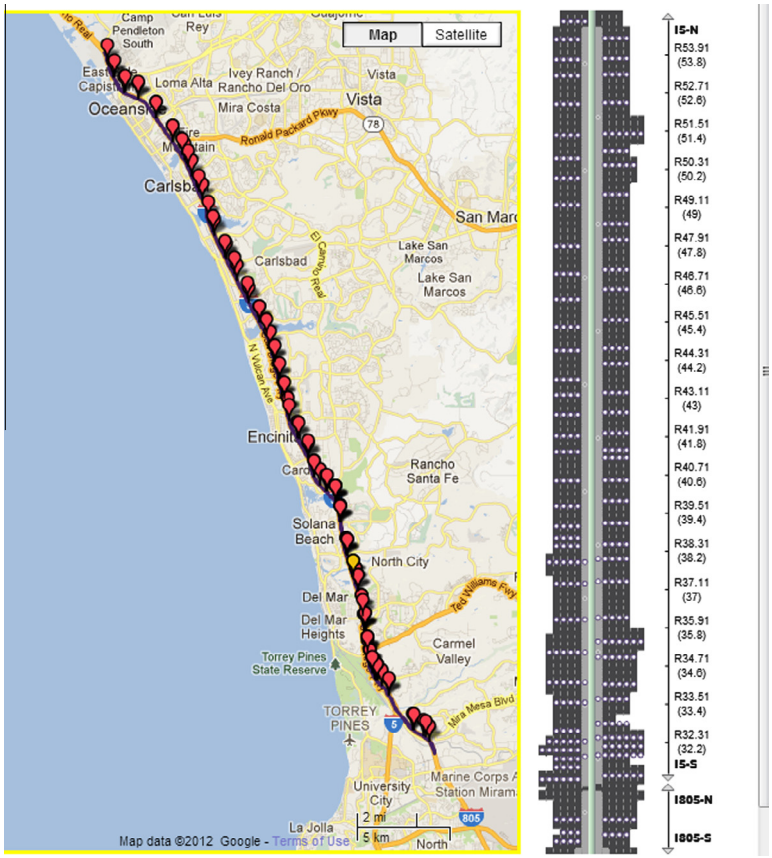

Fig. 4. I-5 corridor in San Diego. *Source*: pems.dot.ca.gov.

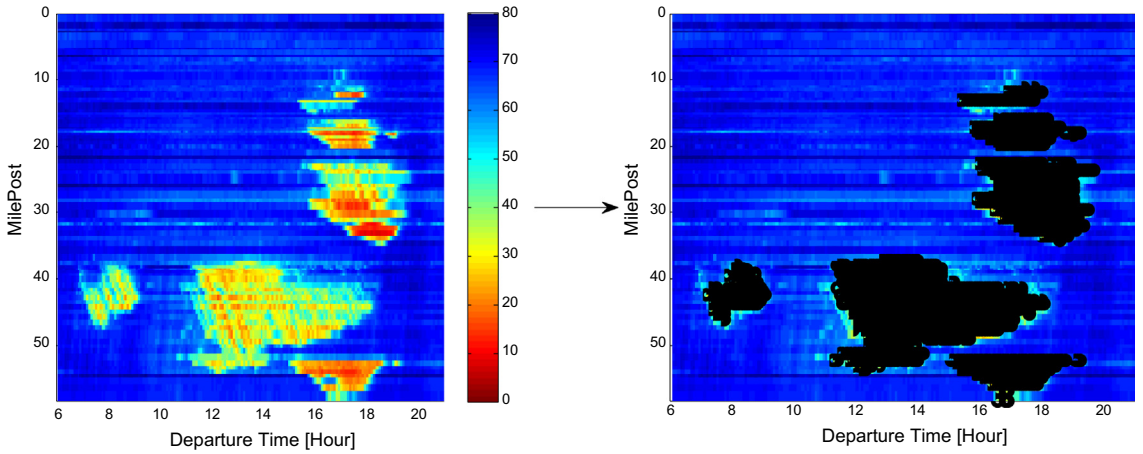

Fig. 5. Identification of congested sections (20-July-2011).

("89 detectors" × "180 5-min time periods" between 6 AM and 9 PM). Therefore, the rest of the clustering operation is carried out with the reduced dataset of 100 variables (components).

Average of silhouette width values in the data set is used, in this study, to determine optimal number of clusters. Fig. 6 presents 100 realizations of clustering for an interval of possible cluster numbers, {2..7}, and the average silhouette width values that result from these realizations. The (vertical) range of average silhouette width values computed for identical cluster numbers clearly shows that results are not stable except three cluster configuration, where the aforementioned range disappears. As the initialization of GMM is random, each run of the algorithm can result in different clusters. Thus, the wide range observed in Fig. 6 implies significant difference between the computed clusters, i.e. instability. In addition, mean average silhouette width (average of 100 realizations) reaches its maximum value at three cluster configuration. Therefore, considering both the stability of the results and the similarity of observations within the cluster (i.e. average silhouette width), optimal number of clusters is selected to be three. We have noticed that by utilizing a higher number of clusters in the methodological framework, the experienced travel time prediction does not improve. Note that a similar analysis can be conducted with AIC and BIC values to determine optimal number of clusters.

Determining the number of clusters is often ambiguous, and it is a separate problem from actually solving the clustering problem. The optimal choice of clusters seeks a balance between the maximum compression of data using a single cluster, and the maximum accuracy by assigning each observation to a separate cluster. Suppose for example that the dataset consists of some dense and distant clusters (e.g. colors), whose number is not known a priori. If we set the number of clusters too low, the algorithm will combine some natural clusters (i.e. different colors) to reduce the total number of groups to the user-specified number of clusters. Additionally, since the initialization of the algorithm is random, each run of the algorithm can result in combination of different natural clusters, which causes instability. On the other hand, if we set the number of clusters too high, then some natural clusters have to be divided in an artificial way, in order to obey the specified number of

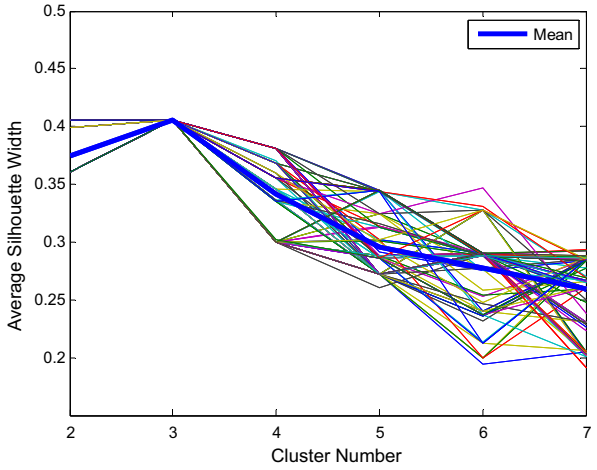

Fig. 6. Average silhouette width vs. cluster number.

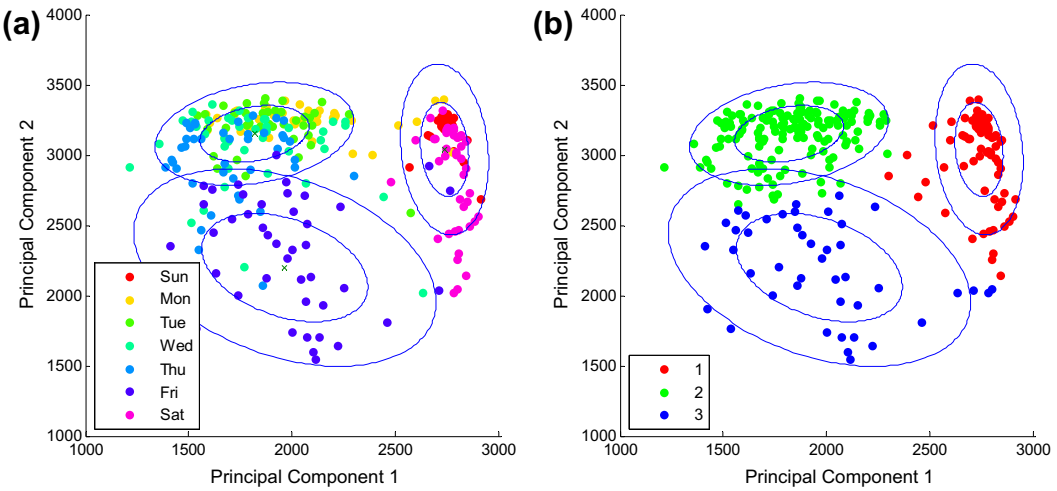

Fig. 7. GMM results (a) based on days of the week and (b) based on clusters.

Table 1
Clusters vs. days of the week.

Cluster	Days						
	Mon	Tue	Wed	Thu	Fri	Sat	Sun
1	6	1	1	1	2	40	42
2	36	40	36	34	8	0	0
3	0	0	3	6	31	2	0

groups, which also causes instability due to the randomness of artificial cuts. Therefore, correct choice of cluster number must be associated with best performance and stability. With respect to our problem, natural clusters may represent uncongested, moderately congested and highly congested traffic conditions. In that case, the clustering algorithm would perform at its optimum with the correct choice of three groups. However, another freeway where natural clusters can be described differently may require a different choice of the optimal number of clusters.

Fig. 7 presents the distribution of the days along the first two principal components and the GMM results. Ellipses in Fig. 7 represent 50% and 90% of the variance of the clusters along the dimensions of the two PC's. Note that the PC values do not have a physical meaning; they represent the values of the new uncorrelated variables. The results clearly show that the clusters are mainly dominated by certain features of days of the week, with some exceptions (about 15% of the days). The first cluster shown in Fig. 7b is dominated by weekend days, the second by week days other than '*Fridays*', and the third by '*Fridays*'.

Table 1 presents the distribution of days along the clusters. Most of the week days classified in the first cluster are holidays that are not subject to a significant level of congestion. The second cluster is mainly composed of week days other than '*Fridays*'. These days have significant level of congestion. However, the level of congestion is not as high as it is observed on the most congested cluster. Therefore, the clustering algorithm creates a separate cluster mainly for '*Fridays*' (25% of this cluster includes days other than Friday). Although the clusters do not totally belong to a certain day of the week, this information is very useful to identify expected traffic conditions on the roadway for a particular day. Hence, each cluster is assigned to dominating days of the week in a predetermined way, and travel time prediction for a given day is executed within the corresponding cluster and its associated congestion map. The switch algorithm will provide the flexibility to switch to a more or less congested cluster if traffic conditions look very different than what was initially assumed.

3.3. Stochastic congestion map

By combining the results obtained from bottleneck identification and clustering steps, stochastic congestion maps can now be created. They can be constructed by simply estimating the average number of congestion observation for a space–time point in each cluster. Fig. 3a and c present the stochastic congestion maps for the third and the first cluster, respectively. Then, they are divided into blocks for different threshold probability values (see Fig. 3b and d).

Note that the shape of blocks in the congestion map may not be totally proper regarding the traffic flow essentials (e.g. shockwave speeds). This may be because of the flaws of the bottleneck identification algorithm or averaging different bottleneck spatio-temporal shapes. However, the purpose of this study is to predict experienced travel times or to predict the

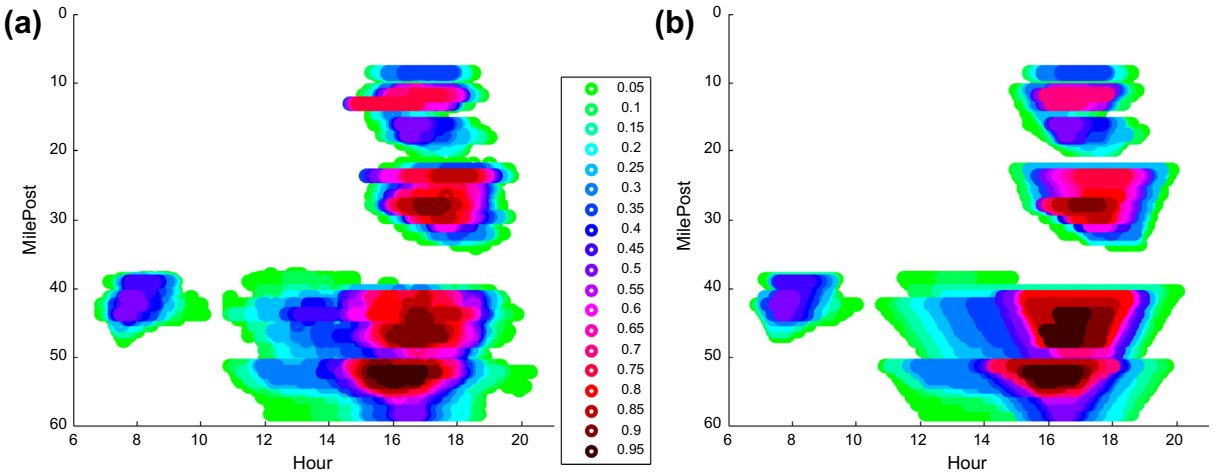

Fig. 8. (a) Original stochastic congestion map and (b) revised stochastic congestion map.

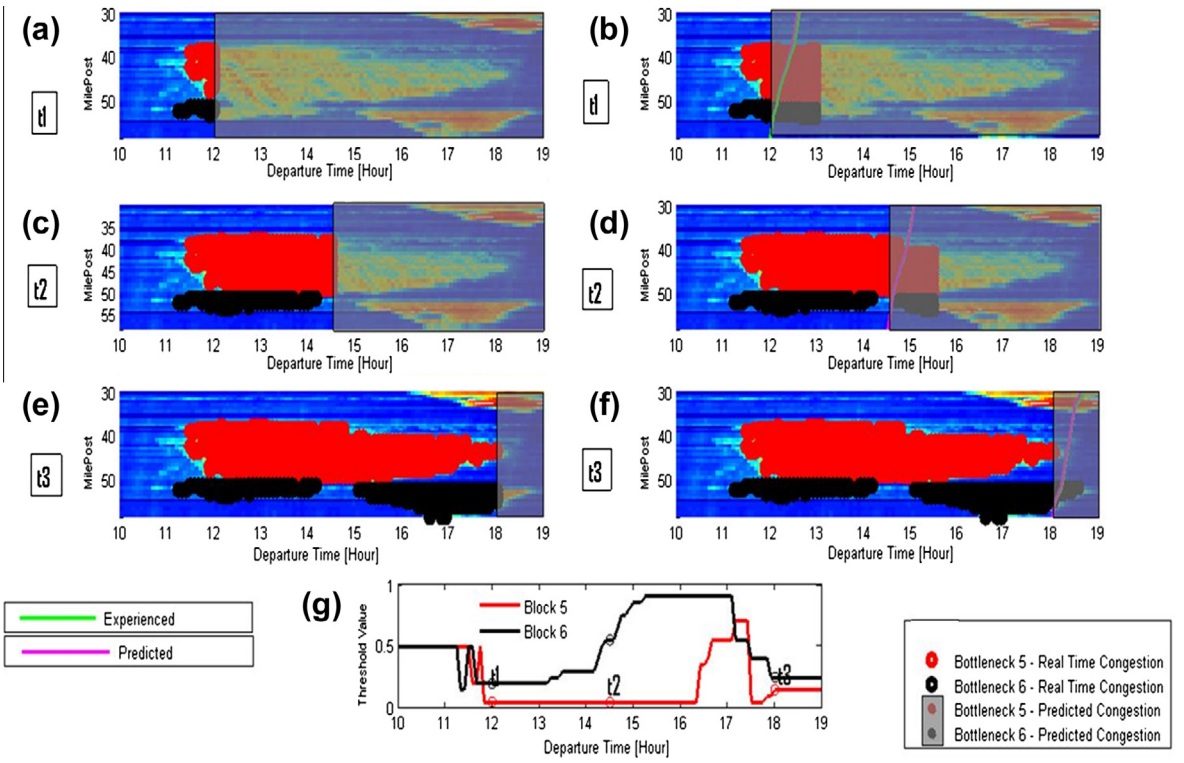

Fig. 9. Graphical representation of the algorithm.

time spent in the queue, not to predict the shape of the bottlenecks. The important feature of an accurate methodology is to estimate with some confidence the number of congested points (in the space–time domain) that the predictive trajectory will intersect while traveling.

To test the influence of bottleneck shapes, stochastic congestion map is revised to incorporate physical bounds on shockwave speeds, and its effects on prediction results are investigated. For each bottleneck and for each subset represented by threshold probability values given in Fig. 8, bottleneck shapes or congestion patterns have been modified to obey the shockwave speed bounds. In this study, maximum shockwave speed is calculated for a triangular fundamental diagram and it is taken as 10 mph. Subset points are fitted, in the best possible way, into an area defined by polylines whose derivative is between 0 and 10 mph. Note that this modification leads to a certain change in the number of congested points defined by the

subset, which is small indeed and does not change the accuracy of the algorithm. Revised congestion map for the 3rd cluster, given in Fig. 8b, provides realistic and accurate bottleneck shapes compared to the original congestion map presented in Fig. 8a. Revised congestion map is evaluated in Section 4.1 as an alternative to the original approach.

3.4. Graphical representation of the algorithm

To further elaborate the implementation of the methodology, Fig. 9 presents the mechanism of the algorithm on a particular day (*Wednesday*, 20-July-2011). Estimation of experienced travel times on this specific day is presented in the next section. Although the pre-assigned cluster for '*Wednesday*'s indicates a moderate level of congestion, this particular day exhibits a higher level of congestion, which is a challenging case for our approach. Non-recurrent congestion on this particular day requires both cluster switch and adjustment of threshold values.

Fig. 9 shows the results of the algorithm for bottlenecks #5 and #6 (refer to Fig. 3b for the index of the bottlenecks). These two bottlenecks have very small size and very low probability in the pre-assigned cluster (see Fig. 3c and d). Therefore, the algorithm switches to the highly congested cluster at 11:15 am for the bottleneck #5, and at 11:30 am for the bottleneck #6. Fig. 9a, c and e show the speed contour plots and the congested points identified by the algorithm until the departure time. Three different departure times are shown, in the onset (t_1 = 12:00), during (t_2 = 14:30) and the offset (t_3 = 18:00) of congestion. Note that although the speed contour plot is given until 19:00, this information is not available at the departure time and the missing portion is shaded with a transparent gray rectangle. Fig. 9b, d and f introduce the same information along with predicted congestion sections within the next 1 h, and experienced and predicted trajectories for the trips that start at the specific departure time. Note that experienced and predicted trajectories are so similar that it is very difficult to distinguish between them especially in Fig. 9d and f.

Fig. 9g introduces the results of the congestion search algorithm; threshold values for bottlenecks #5 and #6. Note that high values of thresholds represent lower levels of congestion, according to the definition in Section 2.3. Note that the output of the online congestion search algorithm is consistent with neighboring time periods. Threshold values are not subject to large variations (up-and-downs), which would represent sudden prediction changes. Fig. 9g also gives the threshold values for the departure time periods t_1, t_2 and t_3. At 12:00 (t_1), bottlenecks are just starting to grow, and the congestion search algorithm chooses very low threshold values for both, considering the very early starting time of the congestion. Note that probability of having congestion at this departure time is very low even in the high congestion cluster (see Fig. 3b). At 14:30 (t_2), the algorithm detects that bottleneck #6 has started to disappear and updates its decision by increasing the corresponding threshold value (from 0.3 to 0.5), while it insists on its decision for bottleneck #5. At 18:00 (t_3), the congestion search algorithm re-identifies congested sections around the bottleneck #6, it decreases its threshold value gradually to adjust the changes in the size and shape of the bottleneck. Note that while identified congested sections in real-time are shown for the whole bottleneck in the figures, a moving time window of the last 2 h prior to the departure time is used in the search mechanism to adjust to rapidly changing traffic conditions.

4. Results

4.1. Evaluation of the proposed approach

The approach presented in this paper (along with original stochastic congestion maps) is evaluated on the testing dataset (76 days). Since the weekend days are not subject to significant level of congestion, they are not considered in the evaluation step. Travel time, in this study, is computed using speed data from loop detectors and constant speed interpolation technique. For a link between two successive detectors, the speed measurement at downstream or upstream detector, or the average of two measurements can be used to represent the velocity. All constant speed interpolation methods imply instantaneous speed changes, which do not occur in real-time. However, considering the distance between the detectors (about 500 m) in our study site, this phenomenon is not expected to largely affect the results. Kothuri et al. (2008) analyze travel time estimation errors that result from midpoint algorithm, and conclude that detector failure is the major cause of high estimation errors. Their analysis revealed that travel time estimates produced by the midpoint algorithm have a good accuracy compared with ground truth probe vehicle runs. Chen et al. (2003), where travel time is calculated from the speed measurements at single detectors, also indicate fair estimation accuracy. Considering the distance between the detectors (about 500 m) and high detector quality in the study site, midpoint algorithm is expected to produce accurate travel time estimates. Travel time can also be computed by using linear and quadratic speed interpolation methods, which do not require instantaneous speed changes. Alternatively, one could apply a more detailed traffic flow model (of first or higher order) to estimate speed between the detectors. Nevertheless, we do not expect the accuracy of the results to improve. Predicted travel time is calculated in the same manner as experienced travel time. However, instead of velocity field, which is unknown at the departure time, predictive trajectory travels through the predicted congestion map and uses an estimated speed profile as described by Eq. (10) to compute the time needed to traverse each segment. Historical travel times are also computed for each day of the week. The median value of the experienced travel times at a given departure time on a given day of the week is taken as historical average value.

To measure the effectiveness of the methods, two statistics, namely mean absolute error (MAE) and mean absolute percentage error (MAPE) are utilized;

$$MAE = \frac{1}{n}\sum_{t=1}^{n}|T(t) - \widehat{T}(t)| \tag{11a}$$

$$MAPE = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{T(t) - \widehat{T}(t)}{T(t) - T_{free}}\right| * 100 \tag{11b}$$

where n is the number of observations (i.e. 5 min long departure time intervals), T_{free} is free-flow travel time, $T(t)$ is experienced travel time computed with Eq. (1b), and $\widehat{T}(t)$ is the travel time provided by the methodology. Note that MAPE provides the percentage error in terms of delay.

Fig. 10c presents MAE values of instantaneous and predictive travel time for 103 congested periods at least 30 min long. Since historical average performs clearly worse than the other methods, it is not shown in Fig. 10c. In 85 out 103 cases, predictive travel time methodology produces better results than instantaneous travel time assumption. Note that the performance of the proposed methodology is significantly better when conditions are more congested. Fig. 10c presents the 103 congested periods irrespective of their lengths. To further elaborate this, congested periods are divided into two groups; periods where MAE is less or more than 2 min (see the rectangles in Fig. 10c). Fig. 10a and b provide the histogram of absolute errors for the two groups (considered as a whole) in a disaggregate way, which implicitly accounts now for the length of congested periods. Departure time intervals (of 5 min) from each group are gathered together, and histogram plots are created using the error values associated with them. Note that although total number of congested periods in two groups is similar, number of departure time intervals (5 min long) within the periods is quite different (as indicated by the difference in total bin counts in Fig. 10a and b), which implies the difference in the length of congested periods in two groups. Average length of congested period is 180 min for MAE < 2 min and 275 min for MAE $\geqslant$ 2 min. Results indicate that short congested periods tend to be less problematic even under instantaneous travel time assumption. However, long congested periods are associated with high error for the instantaneous estimation. Although error distribution in the first group is comparable for instantaneous and predictive travel time (Fig. 10a), our approach outperforms instantaneous one in the congested group (Fig. 10b).

Fig. 11 presents the travel times provided by the proposed methodology, the instantaneous approach (estimated by Eq. (1a)), the experienced travel time (which is considered the ground truth, estimated by Eq. (1b)) and the historical average method for six representative days. It clearly shows that historical average is not capable of producing accurate results under congested conditions. In overall, predictive travel time produces better results during both the onset and offset of the congestion. However, the morning peak in Fig. 11a and the afternoon peaks in Fig. 11e and f show that prediction model has a slightly better performance compared with the instantaneous approach during the congestion onset, while it has a significantly better performance during the offset. As newly available real-time information about the bottleneck becomes available, the predicted value is extremely close to the experienced during the congestion offset. Note that the day presented in Fig. 11d is discussed in details in the graphical representation of the algorithm (Fig. 9).

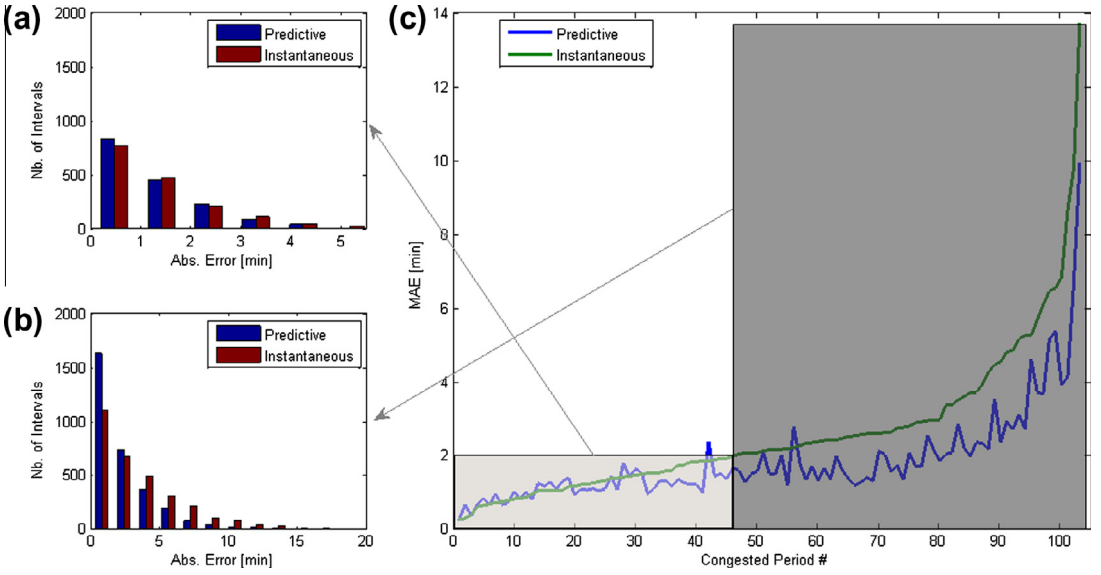

Fig. 10. (a and b) Histogram of absolute errors, and (c) model performance for congested periods.

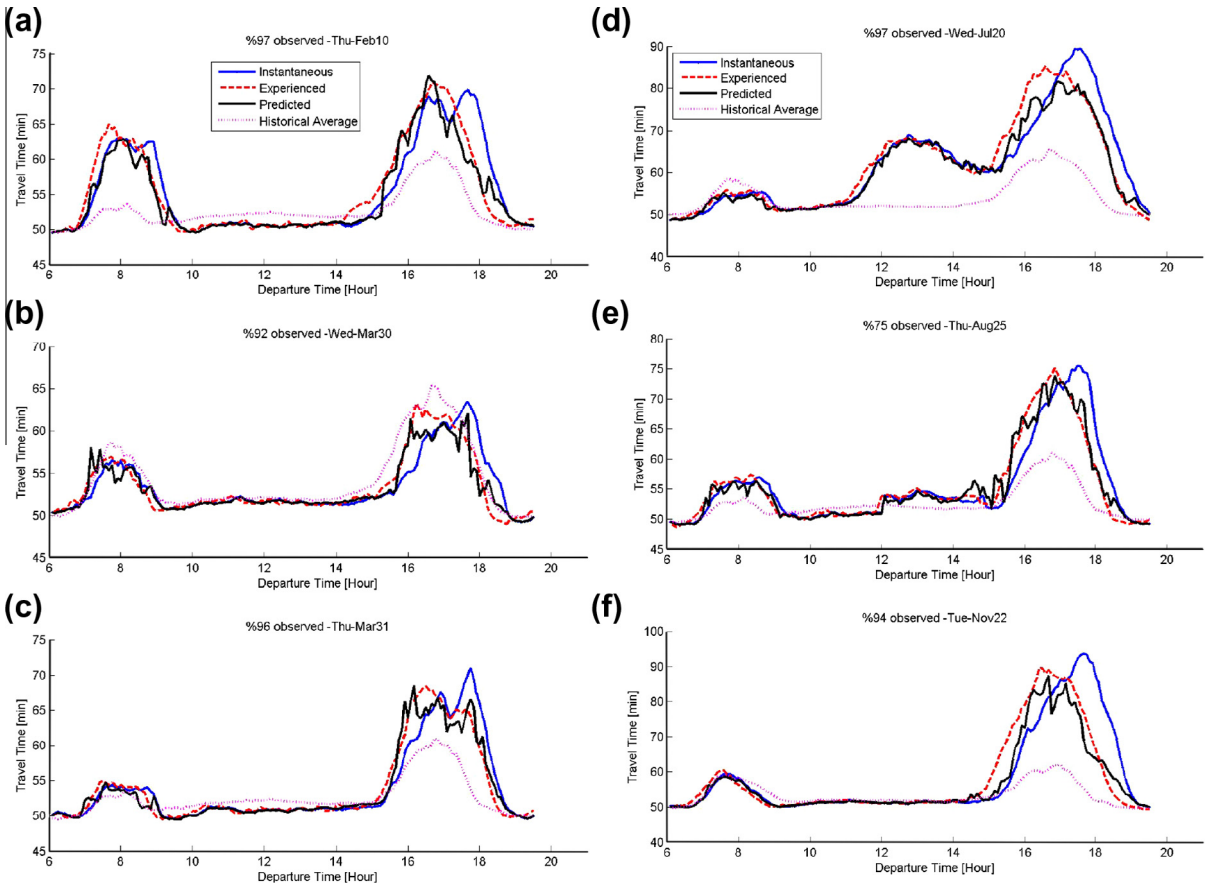

Fig. 11. Travel time series (a) 10-February-2011, (b) 30-March-2011, (c) 31-March-2011, (d) 20-July-2011, (e) 25-August-2011, and (f) 22-November-2011.

Table 2
MAE (min) and MAPE (%) of certain days in the testing set.

Date	Per. ID	Congestion duration	Prediction		Instantaneous		Historical average	
			MAE	MAPE	MAE	MAPE	MAE	MAPE
10-February	63	06:50–09:30 (2:40)	1.98	23	2.43	31	6.85	60
	85	14:10–18:50 (4:40)	1.99	27	3.70	48	5.16	39
30-March	15	06:50–08:55 (1:55)	1.17	18	1.05	18	1.30	22
	58	14:10–18:55 (4:45)	1.34	29	2.34	57	1.68	35
31-March	10	06:55–09:05 (2:10)	1.03	25	0.83	25	1.31	28
	65	14:10–19:00 (4:50)	1.38	18	2.53	30	4.04	32
20-July	13	06:55–09:05 (2:10)	0.81	15	0.97	20	1.60	29
	88	11:05–19:30 (8:25)	2.18	17	4.25	36	12.19	65
25-August	20	06:45–09:20 (2:35)	0.93	17	1.16	24	2.60	38
	74	11:55–19:05 (7:10)	1.35	23	2.77	37	4.74	40
22-November	35	06:45–09:00 (2:15)	1.32	22	1.59	27	2.28	53
	101	13:50–19:15 (5:25)	4.13	34	8.62	76	12.34	51
Weighted avg. of all congested periods in the testing set			2.10	22	3.05	35	5.83	45

Table 2 provides MAE and MAPE values and the congestion duration for the days presented in Fig. 11 along with the period ID that can be matched with Fig. 10c. MAE values indicate that predictive travel time outperforms the other two approaches except for two periods (30-March, morning peak; 31-March, morning peak) where all estimators (even the historical average) have small errors (around 1 min). In addition, Table 2 provides the weighted average of MAE and MAPE values over all the congested periods with respect to their lengths (or durations). Results represent a clear improvement over

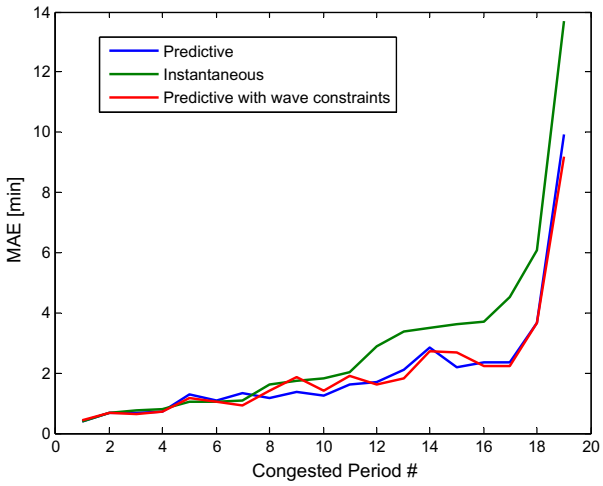

Fig. 12. Comparison of the original and revised stochastic congestion maps.

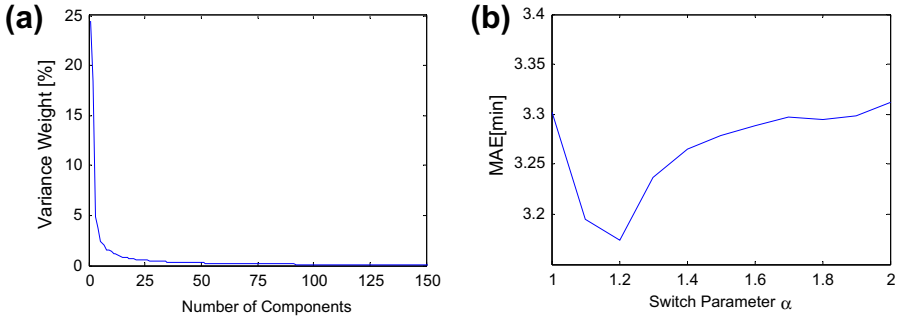

Fig. 13. Sensitivity analysis.

the instantaneous travel time approach; delay percentage error (MAPE) decreases from 35% to 22% (∼40% improvement), and MAE reduces from 3.05 min to 2.10 min (∼30% improvement).

The use of revised congestion map, presented in Fig. 8, instead of its original has allowed us to investigate the influence of bottleneck shapes. Fig. 12 provides evaluation results for 19 congested periods identified in 10 days (mostly 'Fridays') that exist in testing set and that belong to 3rd cluster. Comparison of results reveals no significant improvement. Nevertheless, the motivation of this approach is to provide a methodology which is consistent with the physics of traffic and which can provide a more consistent alternative in case of missing or erroneous data. Although the revised congestion map has accurate bottleneck shapes, time spent in the queue is not significantly affected by this change. This might be the reason of the lack of improvement. Weighted average of MAE values for the periods presented in Fig. 12 is 2.64, 2.70 and 3.90 min, respectively for predictive travel time with shockwave speed constraints, predictive travel time and instantaneous travel time.

4.2. Sensitivity analysis

Parameters used in the prediction framework are (i) the number of clusters (Fig. 6), (ii) the number of components taken from PCA (Fig. 13a) and (iii) the cluster switch parameter, α (Fig. 13b). A sensitivity analysis is conducted to investigate the effect of these parameters on the performance of the proposed prediction model. First, to determine the optimal number of clusters in the dataset, average silhouette width value is computed for a range of possible cluster numbers (Fig. 6). Note that, as the initialization of the algorithm is random, clustering has been implemented several times (100 in this case). As also explained in Section 3.2, two criteria namely stability and performance (i.e. average silhouette width) of results are considered together to determine the optimal number of clusters.

Second, to determine the number of principal components, variance weight along the eigenvector directions (which can be formulated as $\lambda_i/e[n]$ following the notation in Section 2.2.1) is considered. As it is seen in Fig. 13a, components beyond 25th have insignificant contributions to the variance of the data set. However, cumulative variance reaches 95% at 100th component. Hence, the first 100 PCs are chosen to create matrix $\boldsymbol{S}$ (see Eq. (2)). Note that matrix $\boldsymbol{S}$ is utilized only in the

clustering step; the rest of the methodology is applied in the original traffic speed data (matrix $\boldsymbol{X}$ in Eq. (2)). A further analysis has shown that any number of components more than 25 results in identical clusters, which implies stability.

Third, to determine the switch parameter α, days which belong to an alternative cluster instead of their original cluster (e.g. a 'Wednesday' which belongs to 3rd cluster) are identified, and MAE is estimated for different values of α. Fig. 13b presents the weighted average of MAE values for 11 congested periods that comply with the above description. It shows that MAE reaches its minimum at 1.2, which is the optimal value for this study. One can conduct a similar analysis to determine site-specific model parameters.

5. Conclusion

Dissemination of travel time information through ATIS or its use as in ATMS to deploy efficient control measures always requires the prediction of traffic conditions on the freeway. The aim of this paper is to predict travel times by using traffic flow fundamentals, not a pure statistical procedure.

First, an automated bottleneck identification algorithm is applied to detect the major traffic events that occur on the freeway. Then, the historical (or training) dataset is partitioned based on the clusters obtained through GMM. The results obtained from the first two parts are combined to create stochastic congestion maps for each cluster. Next, using the estimated speed profile, the congestion maps associated with threshold values and the congestion search algorithm that connects real-time and historical traffic data, this study predicts the experienced travel times.

In this study, there is no GPS data available. Experienced travel time, which is based on speed measurements at loop detectors and which is presented by Eq. (1b), is used as ground truth travel time. Instantaneous, experienced and predicted travel times are all computed based on piecewise constant speed method. They may exhibit underestimation of travel times. However, the correction would apply to all methods compared. Therefore, it can be considered a systematic bias for all methods used and does not compromise the mutual comparison. This approach could also be compared with data driven approaches. However, this study attempts to produce experienced travel time in the following time interval, while data driven approaches aim to provide instantaneous travel time in future time periods. Hence, it would not be relevant to compare two distinct approaches.

The experiment results based on the loop detector data of I-5S segment in California/San Diego indicate that the proposed method provides promising travel time predictions under varying traffic conditions. This methodological framework could have a great potential to be applied with trajectory data (e.g. GPS devices or smart phones) instead of loop detectors or with a combination of different sensor technologies. Existing advances in bottleneck identification with trajectory data can make this approach easily implementable. An estimation and prediction of travel time distributions for different departure times should also be a research priority, as reliability measures can improve the planning of travel trips for various users and provide tools to traffic management for more efficient control.

References

Adeli, H., 2001. Neural networks in civil engineering: 1989–2000. Computer-Aided Civil and Infrastructure Engineering 16 (2), 126–142.
Chen, C., Petty, K., Skabardonis, A., Varaiya, P., 2001. Freeway performance measurement: mining loop detector data. Transportation Research Record 1748, 96–102.
Chen, C., Skabardonis, A., Varaiya, P., 2003. Travel time reliability as measure of service. Transportation Research Record 1855, 74–79.
Chen, C., Skabardonis, A., Varaiya, P., 2004. Systematic identification of freeway bottlenecks. Transportation Research Record 1867, 46–52.
Coifman, B., 2002. Estimating travel times and vehicle trajectories on freeways using dual loop detectors. Transportation Research Part A 36 (4), 351–364.
Coifman, B., Krishnamurthy, S., 2007. Vehicle re-identification and travel time measurement across freeway junctions using the existing detector infrastructure. Transportation Research Part C 15 (3), 135–153.
Dempster, A.P., Laird, N.M., Rubin, D.B., 1977. Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society Series B (Methodological) 39 (1), 1–38.
Dharia, A., Adeli, H., 2003. Neural network model for rapid forecasting of freeway link travel time. Engineering Applications of Artificial Intelligence 16 (7–8), 607–613.
Dion, F., Rakha, H., 2006. Estimating dynamic roadway travel times using automatic vehicle identification data for low sampling rates. Transportation Research Part B 40 (9), 745–766.
Du, L., Peeta, S., Kim, Y.H., 2012. An adaptive information fusion model to predict the short-term link travel time distribution in dynamic traffic networks. Transportation Research Part B 46 (1), 235–252.
Fei, X., Lu, C.-C., Liu, K., 2011. A Bayesian dynamic linear model approach for real-time short-term freeway travel time prediction. Transportation Research Part C 19 (6), 1306–1318.
Herrera, J.C., Bayen, A.M., 2010. Incorporation of Lagrangian measurements in freeway traffic state estimation. Transportation Research Part B 44 (4), 460–481.
Ji, Y., Geroliminis, N., 2012. On the spatial partitioning of urban transportation networks. Transportation Research Part B 46 (10), 1639–1656.
Jiang, X., Adeli, H., 2004. Wavelet packet-autocorrelation function method for traffic flow pattern analysis. Computer Aided Civil Infrastructure Engineering 19 (5), 324–337.
Kothuri, S.M., Tufte, K.A., Fayed, E., Bertini, R.L., 2008. Toward understanding and reducing errors in real-time estimation of travel times. Transportation Research Record 2049, 21–28.
Leclercq, L., Laval, J.A., Chiabaut, N., 2011. Capacity drops at merges: an endogenous model. Transportation Research Part B 45 (9), 1302–1313.
Ledoux, C., 1997. An urban traffic flow model integrating neural networks. Transportation Research Part C 5 (5), 287–300.
Li, X., Peng, F., Ouyang, Y., 2010. Measurement and estimation of traffic oscillation properties. Transportation Research Part B 44 (1), 1–14.
Liu, Y., Lin, P.W., Lai, X.R., Chang, G.L., Marquess, A., 2006. Developments and applications of simulation-based online travel time prediction system: traveling to Ocean City, Maryland. Transportation Research Record 1959, 92–104.
Mazaré, P.E., Tossavainen, O.P., Bayen, A., Work, D., 2012. Trade-offs between inductive loops and GPS vehicles for travel time estimation: a mobile century case study. In: Transportation Research Board 91st Annual Meeting, Washington, DC.

Milligan, G., 1980. An examination of the effect of six types of error perturbation on fifteen clustering algorithms. Psychometrika 45 (3), 325–342.
Min, W., Wynter, L., 2011. Real-time road traffic prediction with spatio-temporal correlations. Transportation Research Part C 19 (4), 606–616.
Nagendra, S.M.S., Khare, M., 2003. Principal component analysis of urban traffic characteristics and meteorological data. Transportation Research Part D 8 (4), 285–297.
Nanthawichit, C., Nakatsuji, T., Suzuki, H., 2003. Application of probe-vehicle data for real-time traffic-state estimation and short-term travel-time prediction on a freeway. Transportation Research Record 1855, 49–59.
Newell, G.F., 1993. A simplified theory of kinematic waves in highway traffic. Part I, general theory. Transportation Research Part B 27 (4), 281–287.
Okutani, I., Stephanedes, Y.J., 1984. Dynamic prediction of traffic volume through Kalman filtering theory. Transportation Research Part B 18 (1), 1–11.
Rousseeuw, P.J., 1987. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. Computational and Applied Mathematics 20, 53–65.
Treiber, M., Kesting, A., Helbing, D., 2010. Three-phase traffic theory and two-phase models with a fundamental diagram in the light of empirical stylized facts. Transportation Research Part B 44 (8–9), 983–1000.
Van Lint, J.W.C., 2006. Reliable real-time framework for short-term freeway travel time prediction. Journal of Transportation Engineering 132 (12), 921–932.
Van Lint, J.W.C., 2008. Online learning solutions for freeway travel time prediction. IEEE Transactions on Intelligent Transportation Systems 9 (1), 38–47.
Vanajakshi, L., Rilett, L.R., 2007. Support vector machine technique for the short term prediction of travel time. In: IEEE Proceedings of Intelligent Vehicles Symposium, Istanbul, Turkey, pp. 600–605.
Vlahogianni, E.I., Geroliminis, N., Skabardonis, A., 2008. Empirical and analytical investigation of traffic flow regimes and transitions in signalized arterials. Journal of Transportation Engineering 134 (12), 512–522.
Vlahogianni, E.I., Karlaftis, M.G., Golias, J.C., 2005. Optimized and meta-optimized neural networks for short-term traffic flow prediction: a genetic approach. Transportation Research Part C 13 (3), 211–234.
Wang, Y., Papageorgiou, M., Messmer, A., 2008. Real-time freeway traffic state estimation based on extended Kalman filter: adaptive capabilities and real data testing. Transportation Research Part A 42 (10), 1340–1358.
Weijermars, W., Van Berkum, E., 2005. Analyzing highway flow patterns using cluster analysis. In: 8th International IEEE Annual Conference on Intelligent Transportation Systems, Vienna, Austria, pp. 831–836.
Wieczorek, J., Fernández-Moctezuma, R.J., Bertini, R.L., 2010. Techniques for validating an automatic bottleneck detection tool using archived freeway sensor data. Transportation Research Record 2160, 96–106.
Yang, J.-S., 2005. A study of travel time modeling via time series analysis. In: Proceedings of IEEE Conference on Control Applications, Toronto, Canada, pp. 855–860.
Yeon, J., Elefteriadou, L., Lawphongpanich, S., 2008. Travel time estimation on a freeway using discrete time Markov chains. Transportation Research Part B 42 (4), 325–338.
Zhang, X., Rice, J.A., 2003. Short-term travel time prediction. Transportation Research Part C 11 (3–4), 187–210.